In [2]:
```
import pandas as pd
import numpy as np

# Seed for reproducibility
np.random.seed(42)

# Specific geographic hubs in Shivamogga for distance logic
# (Central Hubs: Sahyadri College area, Mc Gann Hospital, City Railway Station)
localities = {
    'Vinoba Nagar': {'rate': 6800, 'school_dist': (0.5, 2.0), 'hosp_dist': (0.5,
    'Gopala': {'rate': 5400, 'school_dist': (1.0, 3.0), 'hosp_dist': (1.5, 4.0)}
    'Alkola': {'rate': 5100, 'school_dist': (1.0, 2.5), 'hosp_dist': (2.0, 4.5)}
    'Bommanakatte': {'rate': 4300, 'school_dist': (2.5, 5.0), 'hosp_dist': (4.0,
    'Basavanagudi': {'rate': 7500, 'school_dist': (0.2, 1.5), 'hosp_dist': (0.2,
    'Sulebailu': {'rate': 3900, 'school_dist': (3.0, 6.0), 'hosp_dist': (5.0, 9.
    'KHB Colony': {'rate': 5800, 'school_dist': (1.0, 2.0), 'hosp_dist': (1.0, 3
    'Gadikoppa': {'rate': 7200, 'school_dist': (0.5, 2.5), 'hosp_dist': (1.0, 3.
}

data = []
for _ in range(2500):
    loc = np.random.choice(list(localities.keys()))
    config = localities[loc]

    # Core Features
    sqft = np.random.randint(600, 4501)
    bhk = 1 if sqft < 900 else (2 if sqft < 1600 else (3 if sqft < 2800 else 4))
    bath = bhk if np.random.rand() > 0.2 else bhk + 1

    # Proximity Features (in Kilometers)
    dist_school = round(np.random.uniform(*config['school_dist']), 2)
    dist_hospital = round(np.random.uniform(*config['hosp_dist']), 2)
    dist_railway = round(np.random.uniform(1.0, 12.0), 2)

    # Amenities (Categorical: 0 or 1)
    parking = 1 if np.random.rand() > 0.2 else 0
    security = 1 if (loc in ['Vinoba Nagar', 'Gadikoppa'] and np.random.rand() >

    # Price Logic: Base + Proximity Penalties + Amenity Bonuses
    price = (sqft * config['rate'])
    price -= (dist_school * 50000) # Prices drop as distance to school increases
    price -= (dist_hospital * 40000)
    price += (parking * 150000) + (security * 200000)

    # Add random market noise
    price += np.random.normal(0, 150000)

    data.append([loc, sqft, bhk, bath, dist_school, dist_hospital, dist_railway,

# Create DataFrame
df = pd.DataFrame(data, columns=[
    'Locality', 'Total_SqFt', 'BHK', 'Bathrooms',
    'Dist_to_School_km', 'Dist_to_Hospital_km', 'Dist_to_Railway_km',
    'Parking_Available', 'Gated_Security', 'Price_INR'
])

# Save and Preview
```

```
df.to_csv('shivamogga_house_data_v2.csv', index=False)
print(df.head())
```

```
      Locality  Total_SqFt  BHK  Bathrooms  Dist_to_School_km  \
0    KHB Colony        4107    4          4               1.73
1     Sulebailu        3980    4          4               5.50
2  Bommanakatte        3985    4          4               3.58
3     Sulebailu        3605    4          4               3.60
4        Alkola        1246    2          3               1.10

   Dist_to_Hospital_km  Dist_to_Railway_km  Parking_Available  Gated_Security  \
0                 2.20                2.72                  0               0
1                 5.85                3.00                  0               0
2                 4.87                7.73                  0               0
3                 7.06                7.52                  0               0
4                 4.37               11.62                  1               0

    Price_INR
0  23883000.0
1  15128000.0
2  16625000.0
3  13385000.0
4   6185000.0
```

In [3]:
```
(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2500 entries, 0 to 2499
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Locality             2500 non-null   object
 1   Total_SqFt           2500 non-null   int64
 2   BHK                  2500 non-null   int64
 3   Bathrooms            2500 non-null   int64
 4   Dist_to_School_km    2500 non-null   float64
 5   Dist_to_Hospital_km  2500 non-null   float64
 6   Dist_to_Railway_km   2500 non-null   float64
 7   Parking_Available    2500 non-null   int64
 8   Gated_Security       2500 non-null   int64
 9   Price_INR            2500 non-null   float64
dtypes: float64(4), int64(5), object(1)
memory usage: 195.4+ KB
```

In [4]:
```
(df.Locality.info())
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 2500 entries, 0 to 2499
Series name: Locality
Non-Null Count  Dtype
--------------  -----
2500 non-null   object
dtypes: object(1)
memory usage: 19.7+ KB
```

In [5]:
```
(df.describe())
```

Out[5]:

	Total_SqFt	BHK	Bathrooms	Dist_to_School_km	Dist_to_Hospital_km
count	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000
mean	2564.398000	3.108800	3.303600	2.100216	3.051136
std	1128.190801	0.955682	1.045688	1.315654	2.092628
min	601.000000	1.000000	1.000000	0.200000	0.200000
25%	1585.500000	2.000000	3.000000	1.180000	1.480000
50%	2561.000000	3.000000	4.000000	1.700000	2.390000
75%	3562.000000	4.000000	4.000000	2.702500	4.192500
max	4498.000000	4.000000	5.000000	5.990000	9.000000

In [6]:
```
df.shape
```

Out[6]:
```
(2500, 10)
```

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Locality              0
Total_SqFt            0
BHK                   0
Bathrooms             0
Dist_to_School_km     0
Dist_to_Hospital_km   0
Dist_to_Railway_km    0
Parking_Available     0
Gated_Security        0
Price_INR             0
dtype: int64
```

In [8]:
```
df['Dist_to_School_km'].fillna(df['Dist_to_School_km'].median(), inplace=True)
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_3052\759060510.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


  df['Dist_to_School_km'].fillna(df['Dist_to_School_km'].median(), inplace=True)
```

In [9]:
```
df.dtypes
```

Out[9]:
```
Locality                object
Total_SqFt               int64
BHK                      int64
Bathrooms                int64
Dist_to_School_km      float64
Dist_to_Hospital_km    float64
Dist_to_Railway_km     float64
Parking_Available        int64
Gated_Security           int64
Price_INR              float64
dtype: object
```

In [10]:
```
# Remove negative or zero prices
df = df[df['Price_INR'] > 0]

# Distance cannot be negative
df = df[(df['Dist_to_School_km'] >= 0) &
        (df['Dist_to_Hospital_km'] >= 0) &
        (df['Dist_to_Railway_km'] >= 0)]
```

In [11]:
```
df.drop_duplicates(inplace=True)
```

In [12]:
```
df['Price_per_SqFt'] = df['Price_INR'] / df['Total_SqFt']
```

In [13]:
```
df['Total_Distance_Score'] = (
    df['Dist_to_School_km'] +
    df['Dist_to_Hospital_km'] +
    df['Dist_to_Railway_km']
)
```

In [14]:
```
df['Bath_per_BHK'] = df['Bathrooms'] / df['BHK']
print()
```

In [15]:
```
df['Amenity_Score'] = df['Parking_Available'] + df['Gated_Security']
```

In [16]:
```
df = pd.get_dummies(df, columns=['Locality'], drop_first=True)
```

In [17]:
```
q1 = df['Price_per_SqFt'].quantile(0.25)
q3 = df['Price_per_SqFt'].quantile(0.75)
iqr = q3 - q1

df = df[(df['Price_per_SqFt'] >= q1 - 1.5 * iqr) &
        (df['Price_per_SqFt'] <= q3 + 1.5 * iqr)]
```

In [18]:
```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

num_cols = [
    'Total_SqFt', 'Dist_to_School_km', 'Dist_to_Hospital_km',
    'Dist_to_Railway_km', 'Price_per_SqFt', 'Total_Distance_Score'
]

df[num_cols] = scaler.fit_transform(df[num_cols])
```

In [19]:
```
df.to_csv('shivamogga_house_data_engineered.csv', index=False)
```

In [20]:
```
# convert all True/False columns into 0 and 1
df[df.select_dtypes(include='bool').columns] = df.select_dtypes(include='bool').
```

In [21]:
```
df.head()
```

Out[21]:

	Total_SqFt	BHK	Bathrooms	Dist_to_School_km	Dist_to_Hospital_km	Dist_to_Railway
0	1.367597	4	4	-0.281449	-0.406812	-1.18
1	1.255005	4	4	2.584619	1.337755	-1.09
2	1.259438	4	4	1.124977	0.869351	0.40
3	0.922548	4	4	1.140182	1.916091	0.34
4	-1.168829	2	3	-0.760395	0.630369	1.64

In [22]:
```
df.to_csv("shivamogga_house_cleaned.csv", index=False)
```